Problema 1 - Cobb-Douglas

1 - a) Enunciado

A seguir está um modelo formulado por Charles Cobb e Paul Douglas, que estabelece a produção (P) em função do capital investido (L) e do trabalho (K).

Sejam $\alpha, b \in \mathbb{R}$, a função g: $\mathbb{R}^2 \to \mathbb{R}$. Considere a notação $P = g(L, K)$.

$$P = bL^{\alpha}K^{1-\alpha}$$

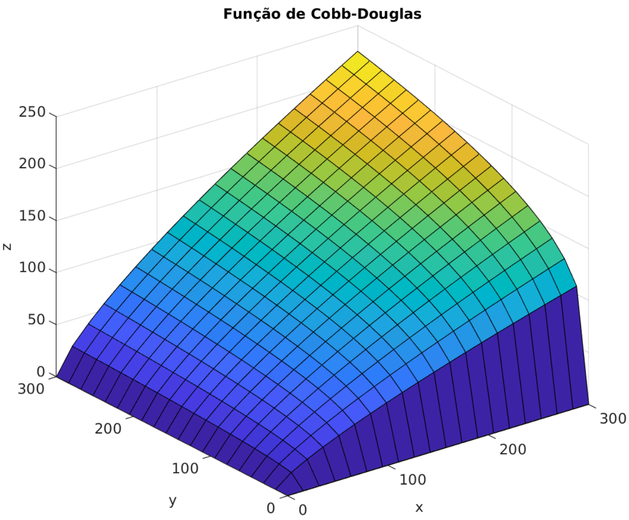
Considere $f : \mathbb{R}^2 \to \mathbb{R}$. Sendo

$$f(L, K) = \ln(b) + \alpha \ln(L) + [1 - \alpha] \ln(K)$$

Com base em um conjunto de dados, queremos obter os parâmetros α e b que melhor se encaixam com esses db) Agora, use a função de Cobb-Douglas encontrada no item a) e teste a sua adequação calculando os valores da produção nos anos de 1910 e
1920. Comente!ados. Para isso, é preciso fazer uma regressão exponencial.

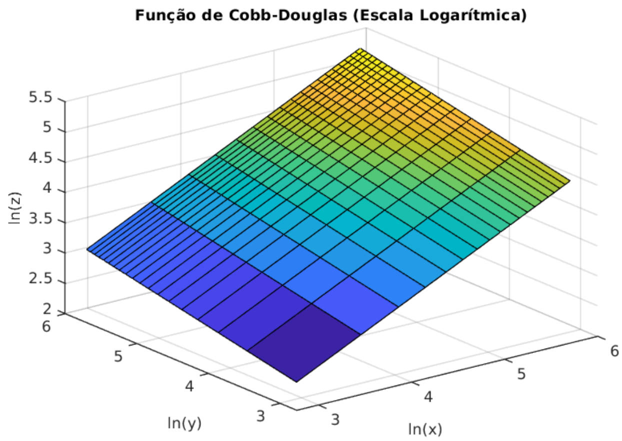
Definida a função, vamos pegar uma restrição dela no nosso conjunto de dados, um conjunto discreto finito. Sejam L_1, K_1, $P_1 \in \mathbb{R}^{24}$. Temos que $P_1 = P|_{L_1 \times K_1}$.

Um técnica para isso, é trabalhar com o conjuntos de dados em escala logarítmica, tornando a superfície que é a imagem da função em um plano. Assim, reduzimos a regressão logísitica a um simple problema de regressão linear sobre um novo conjunto de dados.



$g(x, y)$ com $\alpha = 0.75, b = 0.75$

$P = 0.75\, L^{0.75}\, K^{0.25}$



$f(x, y)$ com $\alpha = 0.75, b = 0.75$

$\ln(P) = 0.75 \ln(L) + 0.25 \ln(K)$

Visto o efeito geométricos dessa mudança de escala, vamos ver a o efeito algébrico, o porquê dessas grandezas passarem a se relacionar de forma linear:

$$\ln(P) = \ln(b) + \alpha \ln(L) + [1 - \alpha] \ln(K) = f(L, K)$$

Aplicamos isso ao nosso conjunto de dados, obtendo os vetores $P_{\ln}$, $L_{\ln}$, $K_{\ln} \in \mathbb{R}^{24}$ que satisfazem

$$P_{\ln} = \ln(b) + \alpha L_{\ln} + [1 - \alpha] K_{\ln} = f|_{L_1 \times K_1}(L, K)$$

Seja $A \in \mathbb{R}^{24 \times 3}$. Podemos expressar a equação acima da seguinte maneira

$$P_{\ln} = \begin{bmatrix} 1 & | & | \\ 1 & | & | \\ 1 & L_{\ln} & K_{\ln} \\ \vdots & | & | \\ 1 & | & | \end{bmatrix} \begin{bmatrix} \ln(b) \\ \alpha \\ 1-\alpha \end{bmatrix} = A \begin{bmatrix} \ln(b) \\ \alpha \\ 1-\alpha \end{bmatrix}$$

Utilizamos o método dos mínimos quadrados para obter os coeficiente α, $\ln(b)$ e $(1-\alpha)$ que descrevem a superfície que mais se aproxima do conjunto de dados. O que consiste em resolver para $\bar{x}$ a seguinte equação

$$A^T A \bar{x} = A^T P_{\ln}$$

Para a resolução dessa equação, usei a função "Gaussian_Elimination_4" que consta na seção de funções extra. Obtendo, idealmente, $(\ln(b), \alpha_1, [1-\alpha_2])$.

O parâmetro b é facilmente obtido pela relação $b = e^{\ln(b)}$. Quanto aos outros parâmetros, percebo que $[\alpha_1 + 1 - \alpha_2] = 1.016$, portanto $\alpha_1 \neq \alpha_2$.

Diante desse impasse quanto ao valor de α, defino uma terceira variável $\alpha_3 = \frac{\alpha_1+\alpha_2}{2}$.

Verifico então qual par (b, α_1), (b, α_2) e (b, α_3) minimiza a soma dos erros ao quadrado, chegando a conclusão que a melhor escolha seja α_2.

Por fim, obtenho os parâmetros que melhor se encaixam aos dados:

$$\alpha = 0.76088\,, \quad b = 0.93313$$

1 - b) Enunciado

Segue

Ano	P	α_1	α_2	α_3
1910	159			
1920	231			

1 - Operações realizadas no MATLAB

```
>> Data = csvread("Datasets/cobb_douglas.csv")
Data =
        1899         100         100         100
        1900         101         105         107
        1901         112         110         114
        1902         122         117         122
        1903         124         122         131
        1904         122         121         138
        1905         143         125         149
        1906         152         134         163
        1907         151         140         176
        1908         126         123         185
        1909         155         143         198
        1910         159         147         208
        1911         153         148         216
        1912         177         155         226
        1913         184         156         236
```

```
        1914        169        152        244
        1915        189        156        266
        1916        225        183        298
        1917        227        198        335
        1918        223        201        366
        1919        218        196        387
        1920        231        194        407
        1921        179        146        417
        1922        240        161        431

>> P = Data(:,2);
>> L = Data(:,3);
>> K = Data(:,4);

>> P_ln = log(P);
>> L_ln = log(L);
>> K_ln = log(K);

>> n = size(P, 1);

>> A = [ones(n, 1), L_ln, K_ln];

>> x = Gaussian_Elimination_4([A' * A], [A' * P_ln])
x =
    -0.069214
      0.76887
      0.24711

>> b = exp(x(1))
b =
      0.93313

>> alpha_1 = x(2);

>> alpha_2 = 1 - x(3)

>> alpha = (alpha_1 + alpha_2)/2
alpha =
      0.76088


>> erro = sum((P - b .* L.^(alpha) .* K.^(1 - alpha)) .^ 2);
>> erro_1 = sum((P - b .* L.^(alpha_1) .* K.^(1 - alpha_1)) .^ 2)
>> erro_2 = sum((P - b .* L.^(alpha_2) .* K.^(1 - alpha_2)) .^ 2)
>> erro_min = sum((P - x(1) .* L.^(x(2)) .* K.^(x(3))) .^ 2)

>> disp([erro,       erro_1,      erro_2,      erro_min])
        7373.6       7815.5       6953.4       2702.8

>> Comparar = zeros(2,4);

>> Comparar(1,1) = P(12);
>> Comparar(2,1) = P(22);
>> Comparar(1,2) = b * L(12)^(alpha_1) * K(12)^(1 - alpha_1);
>> Comparar(1,3) = b * L(12)^(alpha_2) * K(12)^(1 - alpha_2);
>> Comparar(1,4) = b * L(12)^(alpha_3) * K(12)^(1 - alpha_3);
```

```
>> Comparar(2,2) = b * L(22)^(alpha_1) * K(22)^(1 - alpha_1);
>> Comparar(2,3) = b * L(22)^(alpha_2) * K(22)^(1 - alpha_2);
>> Comparar(2,4) = b * L(22)^(alpha_3) * K(22)^(1 - alpha_3);

>> disp(Comparar)
          159       119.46       120.12       119.79
          231       172.68       174.74        173.7


>> Comparar_tudo = zeros(24, 6);

>> Comparar_tudo(:, 1) = Data(:,1);
>> Comparar_tudo(:, 2) = P;
>> Comparar_tudo(:, 3) = b .* L.^(alpha_1) .* K.^(1 - alpha_1);
>> Comparar_tudo(:, 4) = b .* L.^(alpha_2) .* K.^(1 - alpha_2);
>> Comparar_tudo(:, 5) = b .* L.^(alpha_3) .* K.^(1 - alpha_3);
% Valroes da Função que minimiza os erros (deixa de ser a de Cobb-Douglas)
>> Comparar_tudo(:, 6) = b .* L.^(alpha_1) .* K.^(1 - alpha_2);

>> disp(Comparar_tudo)
         1899          100           75           75           75       80.727
         1900          101       79.094       79.118       79.106       85.225
         1901          112       83.184       83.231       83.208       89.723
         1902          122       88.603       88.662       88.632       95.671
         1903          124       93.018       93.124       93.071       100.55
         1904          122        93.55       93.747       93.648       101.21
         1905          143       97.634       97.908       97.771       105.76
         1906          152       105.16       105.49       105.32       114.07
         1907          151        110.7       111.11       110.91       120.24
         1908          126       101.38       102.04       101.71        110.2
         1909          155       115.63       116.23       115.93       125.82
         1910          159       119.46       120.12       119.79       130.09
         1911          153       121.14       121.87        121.5          132
         1912          177       126.84        127.6       127.22       138.31
         1913          184       128.75        129.6       129.17       140.49
         1914          169       127.18       128.14       127.66       138.85
         1915          189       132.36       133.49       132.92       144.71
         1916          225       153.62       154.83       154.22       168.26
         1917          227       167.69       169.11        168.4       184.02
         1918          223       173.15       174.81       173.98       190.27
         1919          218       172.03       173.91       172.97       189.21
         1920          231       172.68       174.74        173.7       190.08
         1921          179       139.56       141.92       140.73       153.68
         1922          240       151.61       154.02       152.81       167.04
```